UCL Université catholique de Louvain

LSINF2224

2014-2015

Programming methods

5.0 crédits ECTS	30.0 h + 15.0 h	2q

Enseignants:	Pecheur Charles ;
Langue d'enseignement:	Anglais
Lieu du cours	Louvain-la-Neuve
Ressources en ligne:	> http://icampus.uclouvain.be/claroline/course/index.php?cid=LINF2224
Thèmes abordés :	-- Fondements de la programmation, leurs propriétés, sémantique, validité et preuve -- Analyse déductives de programme : logique de Hoare, préconditions les plus faibles, conditions de vérification, invariants, et variants. -- Automatisation de preuves : boucles, procédures et récursion, structures de données, programme réactif -- Méthodes d'analyse comportementales : machine d'états, logique temporelle, model checking, abstraction.
Acquis d'apprentissage	Eu égard au référentiel AA du programme « Master ingénieur civil en informatique », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants : -- INFO1.1-3 -- INFO2.5 -- INFO5.3, INFO5.5 -- INFO6.1, INFO6.3 Eu égard au référentiel AA du programme « Master [120] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants : -- SINF1.M3 -- SINF2.5 -- SINF5.3, SINF5.5 -- SINF6.1, SINF6.3 Les étudiants ayant suivi avec fruit ce cours seront capables de -- définir et formaliser les principes de l'analyse et la vérification de introduit dans les cours de baccalauréat. -- décrire et appliquer les techniques qui permettent à ces principes d'être automatisés sur un ordinateur. -- illustrer le potentiel et les limites de ces techniques par des exemples concrets. Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils ont développé leur capacité à -- formaliser sous forme mathématique un problème donné; -- rédiger un rapport technique bref reprend les principaux éléments d'un travail d'analyse; -- argumenter à l'oral. *La contribution de cette UE au développement et à la maîtrise des compétences et acquis du (des) programme(s) est accessible à la fin de cette fiche, dans la partie « Programmes/formations proposant cette unité d'enseignement (UE) ».*

Modes d'évaluation des acquis des étudiants :	-- 3 missions, 45% de la note finale. -- Théorie: examen oral, 55% de la note finale. Une liste de questions est fournie à la fin du quadrimestre. Les missions devront être présentées au cours du quadrimestre de cours. Elles ne pourront pas être représentées lors des sessions d'examens ultérieures.
Méthodes d'enseignement :	Le cours combine -- des cours magistraux, -- des séances de travaux pratiques (modélisation et analyse de programmes), -- et des missions où les étudiants utilisent un logiciel de vérification automatique (ESC / Java, Java PathFinder) afin de prouver des propriétés de programmes Java. Via leurs énoncés, des missions sont clairement balisées. Le résultats attendus, la méthodologie à utiliser, les outils à exploiter y sont présentés. Le code du programme à analyser est anoté et l'énoncé comporte un bref manuel d'utilisation. L'outil de preuve de programme ou de model-checking qui doivent être utilisés sont clairement indiqués. Pour faciliter leur prise en main par les étudiants, des séances d'exercices sont prévues. De plus, une consultance est assurée par les encadrants du cours en cas de problème. Chaque mission fait l'object d'un bref rapport écrit qui servira de base pour l'évaluation.
Contenu :	-- Démonstration -- Introduction -- Fondements -- Programmes séquentiels -- Conditions de vérification -- Procédures -- Récursion -- Structures de données -- Programmes réactifs -- Modèles basés sur des états -- Model Checking -- Abstraction
Bibliographie :	Support de cours : -- transparents en ligne Bibliographie : -- B. Liskov, J. Guttag. Program Development in Java: Abstraction, Specification and Object-Oriented Design. Addison-Wesley, 2001. -- O.-J. Dahl. Verifiable Programming. Prentice Hall, 1992. -- K. R. Apt, E.-R. Olderog. Verification of Sequential and Concurrent Programs. Springer Verlag, 1991. -- J. Loeckx, K. Sieber. The Foundations of Program Verification (2nd Ed.) Wiley-Teubner, 1984. -- D. Gries, The Science of Computer Programming. Springer-Verlag, 1981.
Autres infos :	Préalables: -- LINGI1122
Faculté ou entité en charge:	INFO